

# Improving Human Activity Recognition Through Ranking and Re-ranking

Zhenzhong Lan, Shou-I Yu, Alexander G. Hauptmann  
Carnegie Mellon University

lanzhzh, iyu, alex@cs.cmu.edu

## Abstract

We propose two well-motivated ranking-based methods to enhance the performance of current state-of-the-art human activity recognition systems. First, as an improvement over the classic power normalization method, we propose a parameter-free ranking technique called rank normalization (RaN). RaN normalizes each dimension of the video features to address the sparse and bursty distribution problems of Fisher Vectors and VLAD. Second, inspired by curriculum learning, we introduce a training-free re-ranking technique called multi-class iterative re-ranking (MIR). MIR captures relationships among action classes by separating easy and typical videos from difficult ones and re-ranking the prediction scores of classifiers accordingly. We demonstrate that our methods significantly improve the performance of state-of-the-art motion features on six real-world datasets.

## 1. Introduction

It is a challenging task to recognize human activities among videos, especially unconstrained Internet videos with large visual diversity. A typical human activity recognition pipeline is often composed of the following three steps: (i) extract local video descriptors (e.g., STIP [17], IDT [27] or Deep learning features [28]), (ii) encode and pool the local descriptors into video descriptors (e.g., Fisher Vectors [24] or VLADs [1]), and (iii) classify the video representations (e.g., SVM). The basic components of the current state-of-the-art human activity recognition system, for example, are Improved Dense Trajectories (IDT), Fisher Vectors, and SVMs. In this paper, we introduce two ranking-based methods to improve the performance of this state-of-the-art human activity recognition pipeline.

The first proposed method is to tackle the sparse [24] and bursty [1] distribution problems of Fisher Vectors [24] and VLADs [1]. A common issue of Fisher Vector and VLAD encoding is that they generate high dimensional sparse and bursty video representations whose similarity cannot be accurately assessed by linear measurements. The sparse dis-

tribution, i.e. most of the values in the representation are close to zero, comes from the fact that as the codebook size increases, fewer local descriptors are assigned to each codebook. A bursty distribution, i.e. the values of a few dimensions dominate the representation, occurs when repeated patterns in a video generate a few artificially large components in the representation. To deal with those problems, Perronnin *et al.* [24] invented a simple Power Normalization (PN) method to disperse the video representations. PN uses an element-wise power operation to discount large values and increase small values of video representations. As one of the most significant improvements in the past few years, this simple algorithm essentially makes Fisher Vectors and VLADs useful in practice, and has been widely adopted by the research community to both handcrafted [17, 27] and deeply-learned features [28, 29]. However, PN can only alleviate the sparse and bursty distribution problems. It is often difficult to decide how much dispersal each task requires. There is also no theoretical justification for using sign square root [24, 1] as a rule of thumb. Such being the case, we propose Rank Normalization (RaN) that is parameter-free and thoroughly addresses the sparse bursty distribution problem. RaN ranks all video representations in a dataset along each dimension and uses the normalized rankings in place of the original video representations for the subsequent classification. We show that RaN also works for scenarios where not all data are available. Using an approximation method which requires only a subset of video representations gives a similar ranking performance to the precise case.

The second method we suggest is inspired by curriculum learning which mimics the human learning scheme and has become popular for image and video classification [3, 5, 11, 6]. Curriculum learning [3, 14] suggests distinguishing easy and typical samples from difficult ones and treating them separately. Traditional curriculum learning methods [3, 14, 2] rely on human or extra resources to define data with difficulty levels (curriculum). Instead, we use freely-available classifiers from other action classes to define the curriculum. We then re-rank the prediction results to promote easy videos and suppress difficult ones. Our

proposed method, called Multi-class Iterative Re-ranking (MIR), is easy to implement and training-free.

In the remainder of this paper, we review some relevant works about the improvement of feature encoding, mostly associated with Fisher Vectors and VLADs. We also briefly introduce recent studies on capturing relationships among multiple action classes and curriculum learning. We then describe our two proposed methods in details and demonstrate their performance gain over the baseline approach (composed of IDT, Fisher Vectors, and SVMs) for the action recognition task represented by Hollywood2 and Olympic Sports datasets. Next, we combine those two methods and report results accordingly on both action recognition and event detection [15] tasks. A conclusion and discussion of future works follow in the end.

## 2. Related Work

Features and encoding methods are the major sources of breakthroughs in conventional video representations. Among them the trajectory based approaches [27, 12], especially the Dense Trajectory (DT), IDT [26, 27], and Fisher Vectors encoding, are the basis of current state-of-the-art algorithms.

Fisher Vectors and VLADs are very similar encoding methods [1] and both have been popular for image and video classification [27, 24, 1, 28, 19]. In the original scheme [23, 10] they either do not require post processing [23] or use only L2 normalization [10]. Although L2 normalization can reduce the influence of background information and transform the linear kernel into an L2 similarity measurement [24], it does not disperse the data. As a result, the original Fisher Vector encoding method showed inconclusive results compared to other state-of-the-art encoding methods [23]. It is the introduction of PN [24, 1] that significantly improved the performance of those encoding methods and thus made them useful in practice. PN alleviates the problem of sparse and bursty distribution of Fisher Vector and VLAD. Later on, as a special design for VLAD, Arandjelovic and Zisserman [1] proposed Intra-normalization to further reduce the bursty distribution problem of VLADs. Jégou and Chum [10] used PCA to decorrelate a low dimensional representation and adopted multiple clustering to reduce the quantization errors for VLADs. Nonetheless, those methods only alleviated the burstiness problem. The sparsity of the encoded descriptors still depends on that of the original data. Unlike the above two approaches, RaN normalizes each dimension of Fisher Vectors to a distribution close to the uniform distribution, regardless of how sparse the original data are.

Given the encoded features, state-of-the-art methods often use ‘one versus rest’ SVM, which does not consider the relationships among action classes. To model those relationships, Bergamo & Torresani [4] suggested a meta-class

method for identifying related image classes based on misclassification errors from a validation set. Hou et al. [8] identified similar class pairs and grouped them together to train ‘two versus rest’ classifiers. By combining ‘two versus rest’ with ‘one versus rest’ classifiers, they observed significant improvements from baselines. Unlike the aforementioned approaches that require training and modify the predictions for one time only, MIR is training-free and iteratively updates the prediction rankings given those from previous iterations.

Our MIR model is inspired by a new learning paradigm called curriculum learning, proposed by Bengio et al. [3]. Curriculum learning rates the difficulty levels for classifying samples and uses the rating as a ‘curriculum’ to guide learning. This new way of learning, from a human behavioral perspective, is considered similar to human learning in principle [13]. Moreover, just like school curriculum design in the everyday case, it relies on human or other data resources to define the curriculum. Our method instead uses freely available classifier predictions from other action classes to help define the curriculum without human intervention.

## 3. Action Recognition Preliminaries

**Problem formulation** Given a collection of short clips of videos that usually last a few seconds, the goal of an action recognition task is to classify them into actions of interest such as *running* and *kissing*, solely based on the video content. By evaluating on this task, we dig deep into the characteristics of RaN and MIR.

**Benchmark datasets** We rely on two widely used action recognition benchmark datasets including the Hollywood2 [18] and Olympic Sports datasets [20].

The Hollywood2 dataset [18] contains 12 action classes and 1707 video clips that are collected from 69 different Hollywood movies. There are 12 action classes such as answering a phone, driving a car and standing up. Each video of this dataset may contain multiple actions. We use the clean training dataset and standard splits with training (823 samples) and test videos (884 samples) provided [18]. The performance is evaluated by computing the average precision (AP) for each of the action classes and reporting the mean AP (mAP) over all classes.

The Olympic Sports dataset [20] consists of 16 athletes practicing sports such as high-jump, pole-vault and basketball lay-up. It has a total of 783 video clips. We use standard splits with 649 training clips and 134 test clips and report mAP as in [20] for comparison purposes.

**Experimental settings** We follow the experimental settings in [27]. More specifically, we use IDT features ex-

tracted using 15 frame tracking and camera motion stabilization. PCA is utilized to reduce the dimensionality of IDT descriptors by a factor of two. After reduction, the local descriptors are augmented with three-dimensional normalized location information [16]. Fisher Vector encoding maps the raw descriptors into a Gaussian Mixture Model with 256 Gaussians trained from a set of 256000 randomly sampled data points. Classification is conducted by a ‘one versus rest’ linear SVM classifier with a fixed  $C = 100$  ([27]). For PN, we use sign square root unless otherwise stated.

#### 4. Rank Normalization (RaN)

As discussed in [24, 1], Fisher Vector encoding often generates high dimensional sparse and bursty video representations, whose similarity cannot be accurately quantified by a linear kernel or an L2 similarity measurement. To address this problem, Perronnin *et al.* [24] introduced PN, which has the following element-wise operation:

$$f(z) = \text{sign}(z)|z|^\alpha,$$

where  $0 \leq \alpha \leq 1$ . PN can only alleviate the sparse and bursty distribution problems and it is difficult to determine a good  $\alpha$  for different tasks. To overcome this weakness, we propose RaN, which is parameter-free and handles the bursty distributions in a more fundamental way. RaN applies to each dimension of the Fisher Vectors the following function:

$$f(z) = \text{rank}(z)/N,$$

where  $\text{rank}(z)$  is  $z$ ’s position after sorting along the dimension of all  $N$  Fisher Vectors in the dataset. After RaN, the values in each dimension of Fisher Vectors are spread out and have a distribution close to uniform.

##### 4.1. Qualitative Analysis

To qualitatively characterize the differences between PN and RaN, we visualize their results from different perspectives using all Fisher Vectors in the Hollywood2 dataset. Results are shown in Figure 1, 2, 3, and 4. Figure 1 displays the distributions of Fisher Vector values and their standard deviations along each dimension. As evidenced by Figure 1a and 1b, the distribution of L2-normalized **Fisher Vector values are indeed sparse and bursty**. These sparse and bursty distributed representations would have cosine similarities that clustered around zero (Figure 2a). PN disperses the data and reduces the probability of zero cosine similarities (Figure 2b). However, the distributions after PN are still somewhat sparse (Figure 1c) and bursty (Figure 1d), whose cosine similarities remain centred around zero (Figure 2b). Conversely, RaN disperses the data to an extreme (Figure 1e, 1f) and completely removes zero cosine similarity (Figure 2c).

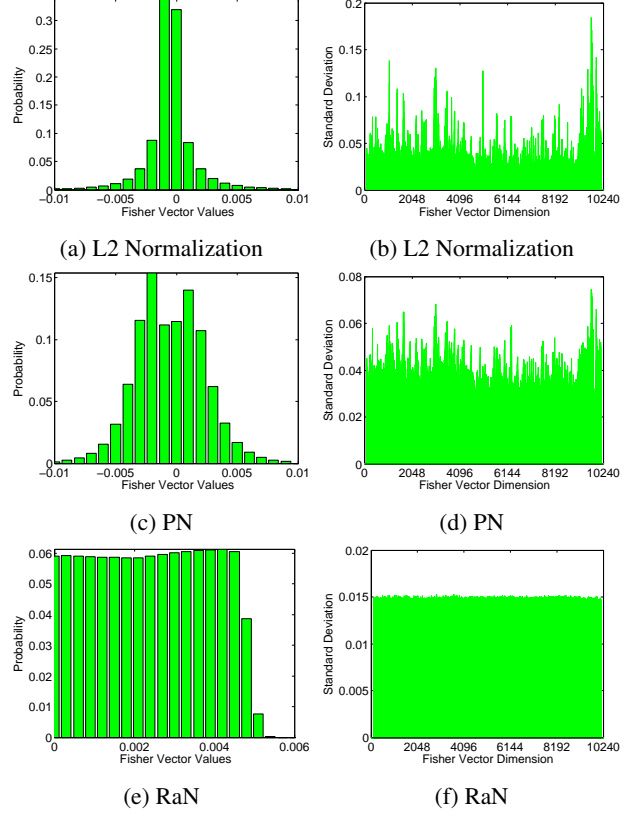


Figure 1: **The effect of different normalization methods on the sparse problem and the bursty problem of Fisher Vectors.** Plots on the left (1a, 1c, 1e) are the distributions of all the values in the Fisher Vectors across all videos for Hollywood2. On the right (1b, 1d, 1f) are the standard deviations of the values in each dimension. L2 normalization serves as a baseline method and is applied after PN and RaN.

In Figure 3, we also compare the normalization effects on the first dimension of Fisher Vectors on the Hollywood2 dataset. The x-axis shows the original values and the y-axis are the corresponding values after normalization. For a better visualization, we normalize all the curves so that the values are between -1 and 1. Basically, PN returns the original values when  $\alpha = 1$  and becomes a step function as  $\alpha = 0$ . When  $0 < \alpha < 1$ , PN carries out a mapping like a sigmoid activation function. It magnifies those values around zeros and allows them to possess more of the y-axis space. Through this magnification, PN assumes that values around zeros are more important than what their original values indicate. However, this assumption provides no information about how much magnification is sufficient. In contrast, RaN states that all Fisher Vector values are equally important in calculating the similarities between video representations.

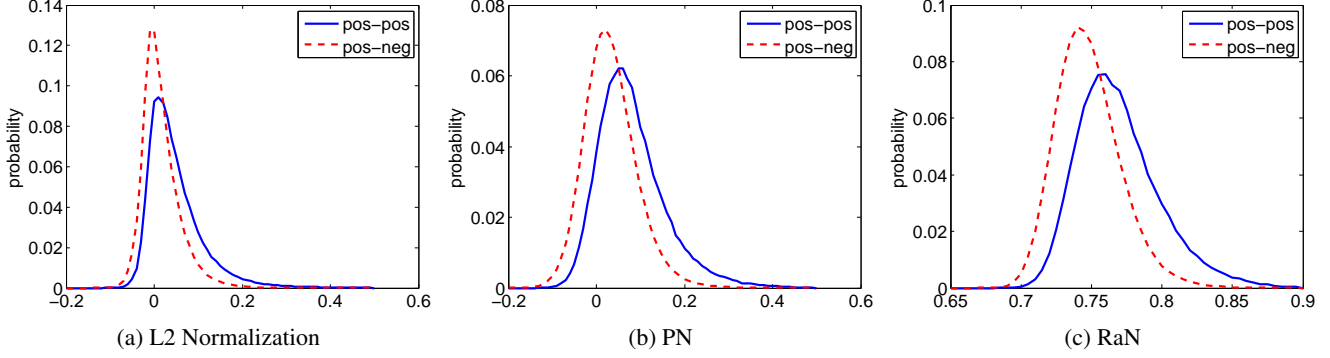


Figure 2: **The cosine similarities of videos under different normalization methods.** The blue and plain lines are the distributions of cosine similarities between positive samples, and the red and dashed lines show the distributions of cosine similarities between positive and negative samples.

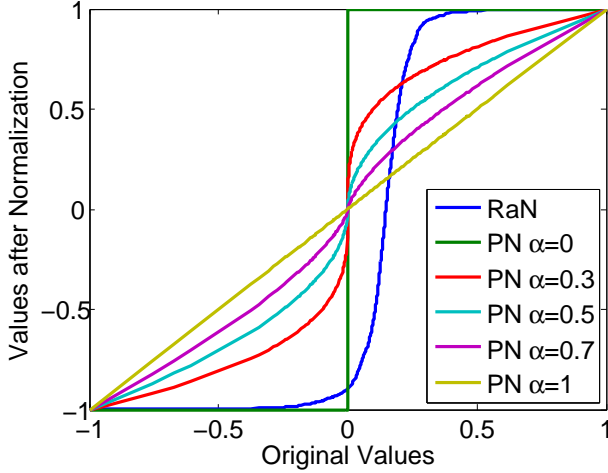


Figure 3: **Comparison of the effects of PN and RaN on the first dimension of Fisher Vectors in Hollywood2.** For a better visualization, we normalize all the curves so that their values are between -1 and 1.

Another interesting observation is that PN cannot dynamically adjust its mapping according to the distribution of values in different sign spaces whereas RaN can. We see from Figure 3 that only a small subset of data are negative; while in the PN case, data with negative values possess as much space as data with positive values. Through dynamic mapping, RaN does much better than PN in spreading out the data (Figure 4). The x-axis of Figure 4 shows the original distribution of L2 distances of the first dimension of Fisher Vectors and the y-axis shows the corresponding distances after normalization. Evidently, after PN, most of the L2 distances are still cluttered around zero; while after RaN, the distribution of L2 distances spread out significantly and become much more separable.

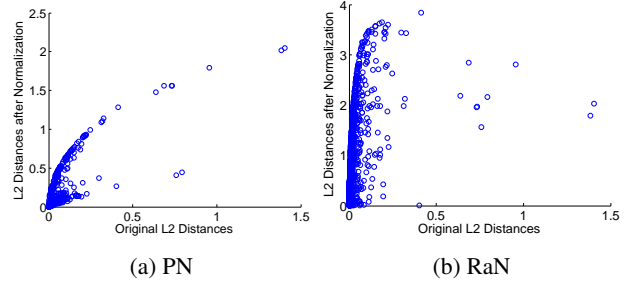


Figure 4: **Comparison of L2 distances before and after normalization on the first dimension of Fisher Vectors in Hollywood2.**

## 4.2. Quantitative Analysis

In Table 1, we compare the results of different normalization methods on both Hollywood2 and Olympic Sports datasets. As in Figure 1, L2 normalization serves as a baseline method and is applied after PN and RaN. First, we show that PN improves the performance of the baseline method on both datasets by more than 6%. These improvements are significant given the difficulty level of the task. RaN further improves the PN performance by more than 1% on Hollywood2 and 3% on Olympic Sports. Compared to the baseline method, RaN gives about 10% absolute improvement on a difficult dataset like Hollywood2; on the relatively easy one like Olympic Sports that has less potential to explore, RaN still manages to improve by more than 9%, absolutely. We also find that applying RaN to local descriptors within each video and combining the rankings with the original descriptors (Two-level RaN) can further improve the performance of Hollywood2 by about 1%, reaching an overall improvement of around 11%. We hypothesize that this is related to removing background distribution as suggested in [30], but here we do not further investigate this connec-

	Hollywood2	Olympic Sports
L2 Normalization	57.9%	83.0%
PN	66.1%	89.4%
RaN	67.7%	92.3%
Two-level RaN	<b>68.6%</b>	<b>92.5%</b>

Table 1: **Performance comparison of different normalization methods on action recognition datasets.** L2 normalization is the baseline method and has also been applied after PN and RaN.

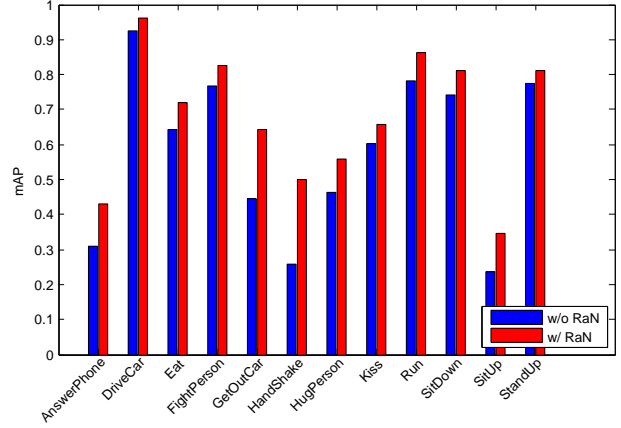
S	Hollywood2 (%)	Olympic Sports (%)
1	43.8 $\pm$ 24.0	31.6 $\pm$ 27.5
5	67.6 $\pm$ 0.2	92.1 $\pm$ 0.9
10	67.5 $\pm$ 0.3	92.6 $\pm$ 0.4
50	67.6 $\pm$ 0.1	92.7 $\pm$ 0.2
100	67.7 $\pm$ 0.1	92.7 $\pm$ 0.1

Table 2: **Comparison of different subset size S for RaN.** Each experiment is repeated 10 times and the mean values and standard deviations are shown.

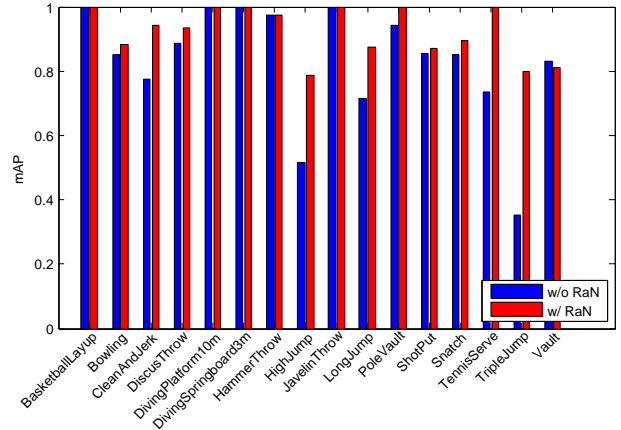
tion. Figure 5 shows the per-class comparison of with and without RaN. Most remarkably, RaN improves the baseline method on all 12 actions from the Hollywood2 dataset. On some of the hard classes like ‘HandShake’, RaN improves the baseline results by more than 15%. A similar trend can be seen in the Olympic Sports dataset. Compared to the baseline method, RaN either improves or delivers similar results on 15 out of 16 actions. These per-class performance comparisons show that our improvements are robust and significant.

### 4.3. Approximate Ranking

The exact ranking of RaN requires comparing all the Fisher Vectors in a dataset, which may not be available in many application scenarios such as online learning. Given the high dimensionality of the data, we conjecture that exact ranking may not be necessary. To test our hypothesis, we randomly choose a small subset of Fisher Vectors with size S as seed vectors and compare each Fisher Vector with the seed vectors to get an approximate ranking. S=1 is equivalent to a binarization of the Fisher Vectors. We experiment with different S on Hollywood2 and Olympic Sports datasets and repeat each experiment 10 times. The results are shown in Table 2. Surprisingly, a subset size S as small as 5 is good enough to achieve similar results as the precise ranking. We also observe better performance from Olympic Sports after using approximate ranking. These results show that fine-grain ranking information is not necessary for RaN.



(a) Hollywood2



(b) Olympic Sports

Figure 5: **Per-class performance comparison of the baseline performances with and without RaN.**

## 5. Multi-class Iterative Re-ranking (MIR)

As mentioned previously, traditional curriculum learning [3] often relies on human or extra data sources to define the curriculum: easy and typical samples versus difficult ones. In this paper, we instead rely on classifiers of other classes, which are freely available, to define the curriculum. The new way of curriculum definition captures relationship among multiple human activity classes.

As depicted in Figure 6, we rank the videos from easy to difficult based on the classifier values. Obviously, easier videos have a much faster roll-off rate of sorted classification predictions. According to this ranking, we discover that videos that have some combinations of the following three scenarios will be more likely to be ranked on the difficult end of the scale:

- Contains noisy background motions. If the background of a video contains noisy motions and the tar-





Figure 6: **Illustration of easy and typical videos versus difficult ones.** The bar charts show the sorted predictions of all the classifiers to the example videos. The predictions are normalized so that the values are between 0 and 1. As shown, the predictions of typical easy videos often have one or two dominant classes while those predictions of difficult videos often have much more smooth score distributions.

---

**Algorithm 1** Multi-class Iterative Re-ranking (MIR)

---

- 1: **Input:** The prediction scores of  $K$  class  $P \in \mathbb{R}^{N \times K}$ ; Re-ranking annealing parameter  $\eta$ ; Re-ranking weighting coefficient  $\beta > 0$ ; Total iteration steps  $W$ .
- 2: **Init:**  $P^{(0)} = P$
- 3: **for**  $w = 1, 2, \dots, W - 1$  **do**
- 4:   For any instance index  $i \in \{1, 2, \dots, N\}$  and class index  $j \in \{1, 2, \dots, K\}$ ,

$$\Delta_{i,j}^{(w)} = \text{sort}(\{P_{i,1}^{(w)}, P_{i,2}^{(w)}, \dots, P_{i,K}^{(w)}\} \setminus P_{i,j}^{(w)}, \downarrow)$$

$$P_{i,j}^{(w+1)} = P_{i,j}^{(w)} - \eta^{w-1} \sum_{\substack{r=1 \\ r \neq j}}^K e^{-\beta r} \Delta_{i,j}^{(w)}(r)$$

- 5: **end for**
  - 6: **Output:**  $P^{(W)}$
- 

get action has a small and weak signal, then the target action would be obscured. For example, the videos of HandShake actions in 6b are obscured by the background motions and much more difficult to detect than those videos of HandShake actions in 6a.

- Contains multiple actions. If the subject performs multiple actions, the classifiers would be confused as the video features represent a mixture of the multiple actions. For example, the Kiss video in 6b contain both Kiss and Hug actions (appear in the following frames) while the Kiss examples in 6a only contain the Kiss action itself.

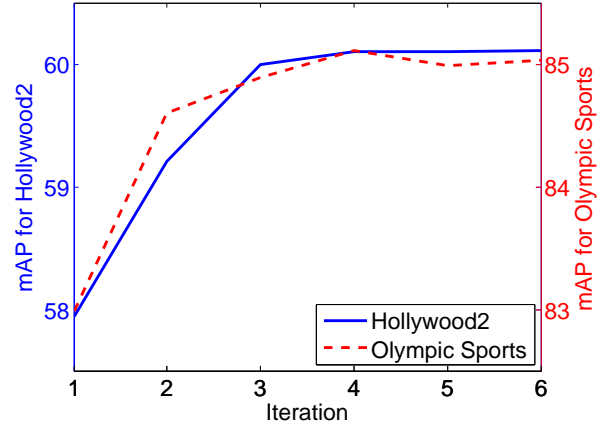
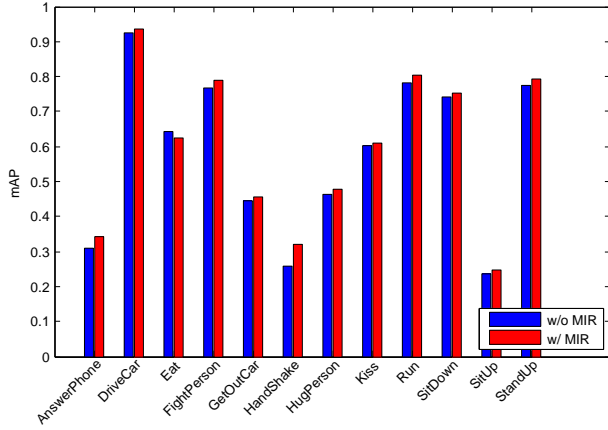


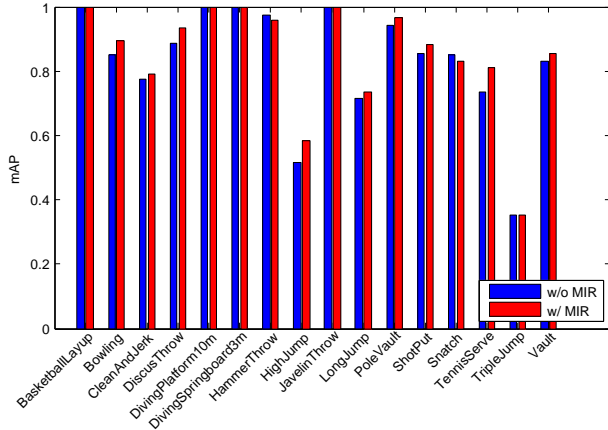
Figure 7: **MIR performance versus number of iterations.**

- Ill-defined actions. If the target action is ill-defined by itself, the video would be more difficult to classify. For example, 'SitUp' action is often encapsulated in the 'StandUp' action, which make it harder to classify.

Based on the difficulty scale, we design MIR to re-rank the predictions of classifiers. The intuition behind MIR is that if videos are more difficult to classify, then their predictions are less reliable and their rank should be lowered; if videos are easy and typical, then the predictions should be more reliable and the videos should be ranked higher. The algorithm to re-rank is easy to implement and fast to run. As shown in Algorithm 1, given a score matrix  $P \in \mathbb{R}^{N \times K}$  that contains  $K$  classifiers' predictions on  $N$  videos. We update each score  $P_{i,j}$  iteratively by looking at other classi-



(a) Hollywood2



(b) Olympic Sports

Figure 8: Per-class performance comparison of the baseline performances with and without MIR.

fiers’ predictions on the same video and reducing the score using the predictions from the other classifiers. The reduction is carried out by first sorting other classifiers’ predictions  $\{P_{i,1}^{(w)}, P_{i,2}^{(w)}, \dots, P_{i,K}^{(w)}\} \setminus P_{i,j}^{(w)}$  in a descending order and then subtracting the weighted sum of the sorted scores from  $P_{i,j}$ . We use exponentially decaying weights and the weighting coefficient  $\beta$  and the annealing parameter  $\eta$  have been set to 1 and 0.5, respectively, throughout the paper. As shown in Figure 7, MIR typically converges within 3 or 4 iterations. It improves more than 2% over the baseline method on both datasets. We also show the per-class comparison of with and without MIR in Figure 8. We can see that for Hollywood2, MIR improves upon the baseline results on 11 out of 12 actions, and for Olympic Sports, MIR improves or gets similar results on 14 out of 16 actions. Different from the improvements of RaN, the improvements of MIR are more evenly distributed among action classes. These per-class performance comparisons again show that

Hollywood2 (%)		Olympics Sports (%)	
Sapienz <i>et al.</i> [25]	59.6	Jain <i>et al.</i> [9]	83.2
Jain <i>et al.</i> [9]	62.5	Oneata <i>et al.</i> [21]	84.6
Oneata <i>et al.</i> [21]	63.3	Adrien <i>et al.</i> [7]	85.5
Wang <i>et al.</i> [27]	64.3	Wang <i>et al.</i> [27]	91.1
Lan <i>et al.</i> [16]	68.0	Lan <i>et al.</i> [16]	91.4
Combined	<b>71.4</b>	Combined	<b>93.6</b>

Table 3: Comparison of our results to the state-of-the-arts on the action recognition task. ‘Combined’ indicates applying both RaN and MIR to the baseline method.

the improvements from MIR are robust and significant.

## 6. Combined Results: RaN and MIR

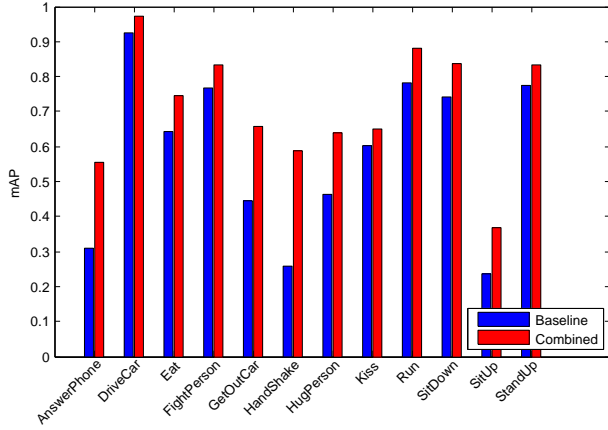
### 6.1. Action Recognition

If we combine both proposed methods, we observe an even more prominent improvement over the baseline method. We improve upon the baseline method by more than 13% and 10% absolute performance improvement on Hollywood2 and Olympic Sports datasets, respectively. A more detailed comparison is provided in Figure 9, from which we can see that our proposed methods together improve the baseline method on all the action classes in Hollywood2. For some of the hard classes like ‘Hand Shake’ and ‘Answer Phone’, we can get more than 20% absolute improvement. For Olympic Sports dataset, we observe a similar trend and get 9 out of 16 classes with perfect predictions.

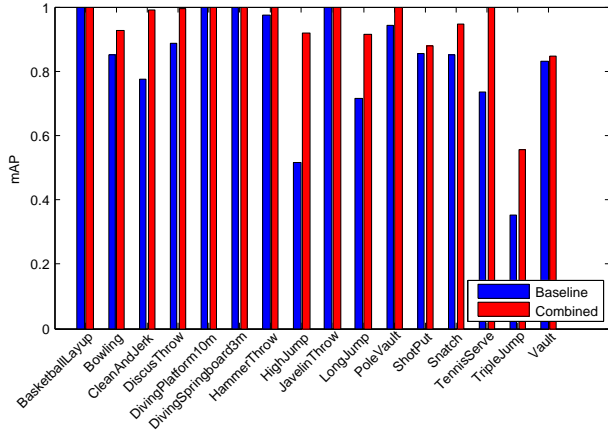
In Table 3, we compare our combined results to the state-of-the-art performance on Hollywood2 and Olympic Sports datasets. Note that although we list several most recent approaches here for comparison purposes, most of them are not directly comparable to our results due to the use of different features and representations. The most comparable one is Wang & Schmid [27], from which we build our approaches on. Sapienz *et al.* [25] explored ways to sub-sample and generate vocabularies for Dense Trajectory features. Jain *et al.* [9]’s approach incorporated a new motion descriptor. Oneata *et al.* [21] focused on testing Spatial Fisher Vector for multiple actions and event tasks. Adrien *et al.* [7] tried to cluster low-level features into mid-level representations in a hierarchical way. The learned hierarchies of mid-level motion components are data-driven decompositions specific to each video. As can be seen, our combined method significantly outperforms these state-of-the-art methods.

### 6.2. Multimedia Event Detection

**Problem formulation** Given a collection of videos, the goal of an event detection task is to detect events of interest such as birthday party and parade, solely based on the video



(a) Hollywood2



(b) Olympic Sports

Figure 9: **Per-class performance comparison of our combined results to the baseline method.** ‘Combined’ indicates applying both RaN and MIR to the baseline method.

content. The task is challenging due to complex actions and scenes. By evaluating on this task, we examine how RaN and MIR behavior on difficult event detection tasks.

**Benchmark datasets** TREC Video Retrieval Evaluation (TRECVID) Multimedia Event Detection (MED) [22] is a task organized by NIST (National Institute of Standards and Technology) aimed at encouraging new technologies for detecting complex events such as *having a birthday party* and *rock climbing*. Until 2014, NIST has built up a database that contains 8000 hours of videos and 40 events, which is by far the largest human labeled event detection collection. MEDTEST13, 14 are two standard evaluation datasets released by NIST in 2013 and 2014, respectively. Each of them contains around 10 percent of the whole MED collection and has 20 events and consist of two tasks, i.e. EK100 and EK10. EK100 has 100 positive training samples per

	MEDTEST13		MEDTEST14	
	EK10	EK100	EK10	EK100
Baseline	17.0	33.6	12.0	26.2
RaN	<b>20.2</b>	36.6	<b>15.4</b>	29.3
MIR	16.7	34.2	11.3	26.6
Combined	20.0	<b>37.5</b>	14.9	<b>29.9</b>

Table 4: **Performance Comparison on the MED task.**

event while EK10 only has 10. Together, each dataset has 8000 training samples and 24000 testing samples.

**Experimental settings** This is similar to the settings discussed in section 3.

**Results** Table 4 lists the overall mAP on all four datasets. The baseline method is a conventional IDT representation. First, on all four datasets, RaN notably improves the performance of the conventional IDT representation. These results demonstrate that RaN is robust across tasks with different difficulty levels. For MIR, in EK10 scenario where the baseline performance is unusually low, MIR hurts the performance due to the inaccurate curriculum estimation; in EK100 settings that have comparatively reasonable baseline performance, MIR manages to achieve noticeable improvements, though not as much as the improvements in action recognition tasks where the baseline performances are much higher. These results reveal the fact that for MIR being useful, the baseline performance should be reasonably accurate. Finally, the combined results, though not always the best, improve upon the baseline method by around 4% on EK100 tasks and about 3% on EK10 tasks. It is worth emphasizing that MED is such a challenging task that 3% of absolute performance improvement is significant.

## 7. Conclusions and Discussions

This paper has introduced two ranking-based methods to improve the performance of state-of-the-art action recognition systems. RaN ranks each dimension of Fisher Vectors and uses the ranks to replace the original vectors. It improves the classic PN method by addressing the sparse and bursty distribution problem of Fisher Vectors and VLADs. MIR iteratively uses the predictions of other classifiers to rank videos from easy to difficult task levels and re-ranks the predictions accordingly. These two methods together significantly improve the performance of Fisher Vectors on six real-world datasets and set new state-of-the-art results for two benchmark action datasets including Hollywood2 and Olympic Sports. Future work will be investigating how to use RaN for local descriptors. In addition, we would like to apply the proposed methods to VLAD and other video features such as deep neural network features.



## References

- [1] R. Arandjelovic and A. Zisserman. All about vlad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2, 3
- [2] Y. Bengio. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009. 1
- [3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, 2009. 1, 2, 5
- [4] A. Bergamo and L. Torresani. Meta-class features for large-scale object categorization on a budget. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2
- [5] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. *arXiv preprint arXiv:1505.01554*, 2015. 1
- [6] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1
- [7] A. Gaidon, Z. Harchaoui, and C. Schmid. Activity representation with motion hierarchies. *International Journal of Computer Vision (IJCV)*, 107(3):219–238, 2014. 7
- [8] R. Hou, A. R. Zamir, R. Sukthankar, and M. Shah. Damn-discriminative and mutually nearest: Exploiting pairwise category proximity for video action recognition. In *Proceedings of European Conference on Computer Vision (ECCV)*. 2014. 2
- [9] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 7
- [10] H. Jégou and O. Chum. Negative evidences and co-occurrences in image retrieval: The benefit of pca and whitening. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 774–787. Springer, 2012. 2
- [11] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann. Self-paced curriculum learning. In *Proceedings of Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, 2015. 1
- [12] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In *Proceedings of European Conference on Computer Vision (ECCV)*. 2012. 2
- [13] F. Khan, B. Mutlu, and X. Zhu. How do humans teach: On curriculum learning and teaching dimension. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1449–1457, 2011. 2
- [14] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 1189–1197, 2010. 1
- [15] Z. Lan, L. Jiang, S.-I. Yu, S. Rawat, Y. Cai, C. Gao, S. Xu, H. Shen, X. Li, Y. Wang, et al. Cmu-informedia at trecvid 2013 multimedia event detection. In *Proceedings of the TRECVID 2013 Workshop*, volume 1, page 5, 2013. 2
- [16] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3, 7
- [17] I. Laptev. On space-time interest points. *International Journal of Computer Vision (IJCV)*, 64(2-3):107–123, 2005. 1
- [18] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2
- [19] I. Mironică, B. Ionescu, J. Uijlings, and N. Sebe. Fisher kernel temporal variation-based relevance feedback for video retrieval. *Computer Vision and Image Understanding*, 2015. 2
- [20] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of European Conference on Computer Visions*. 2010. 2
- [21] D. Oneata, J. Verbeek, C. Schmid, et al. Action and event recognition with fisher vectors on a compact feature set. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (ICCV)*, 2013. 7
- [22] P. Over, G. Awad, J. Fiscus, and G. Sanders. Trecvid 2013—an introduction to the goals, tasks, data, evaluation mechanisms, and metrics. 2013. 8
- [23] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 2
- [24] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the on European Conferences on Computer Vision (ECCV)*. 2010. 1, 2, 3
- [25] M. Sapienza, F. Cuzzolin, and P. H. Torr. Feature sampling and partitioning for visual vocabulary generation on large action classification datasets. *arXiv preprint arXiv:1405.7545*, 2014. 7
- [26] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 2
- [27] H. Wang, C. Schmid, et al. Action recognition with improved trajectories. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (ICCV)*, 2013. 1, 2, 3, 7
- [28] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. *arXiv preprint arXiv:1411.4006*, 2014. 1, 2
- [29] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained cnn architectures for unconstrained video classification. *arXiv preprint arXiv:1503.04144*, 2015. 1
- [30] X. Zhang, Z. Li, L. Zhang, W.-Y. Ma, and H.-Y. Shum. Efficient indexing for large scale visual search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1103–1110. IEEE, 2009. 4